

---

# Guitar Tuner

21<sup>st</sup> February 2017

## OVERVIEW

A guitar tuner is an application that responds to sound generated by a musical instrument and outputs the note the instrument is playing. Stringed instruments have a tendency to loosen due to constant application of tension in the strings, and thus require tuning quite often.

## SPECIFICATIONS

All musical notes have a specific fundamental frequency, and further harmonics are integral multiples of the same, as in the figure.

	C	C#	D	E $\flat$	E	F	F#	G	G#	A	B $\flat$	B
0	16.35	17.32	18.35	19.45	20.60	21.83	23.12	24.50	25.96	27.50	29.14	30.87
1	32.70	34.65	36.71	38.89	41.20	43.65	46.25	49.00	51.91	55.00	58.27	61.74
2	65.41	69.30	73.42	77.78	82.41	87.31	92.50	98.00	103.8	110.0	116.5	123.5
3	130.8	138.6	146.8	155.6	164.8	174.6	185.0	196.0	207.7	220.0	233.1	246.9
4	261.6	277.2	293.7	311.1	329.6	349.2	370.0	392.0	415.3	440.0	466.2	493.9
5	523.3	554.4	587.3	622.3	659.3	698.5	740.0	784.0	830.6	880.0	932.3	987.8
6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	1976
7	2093	2217	2349	2489	2637	2794	2960	3136	3322	3520	3729	3951
8	4186	4435	4699	4978	5274	5588	5920	6272	6645	7040	7459	7902

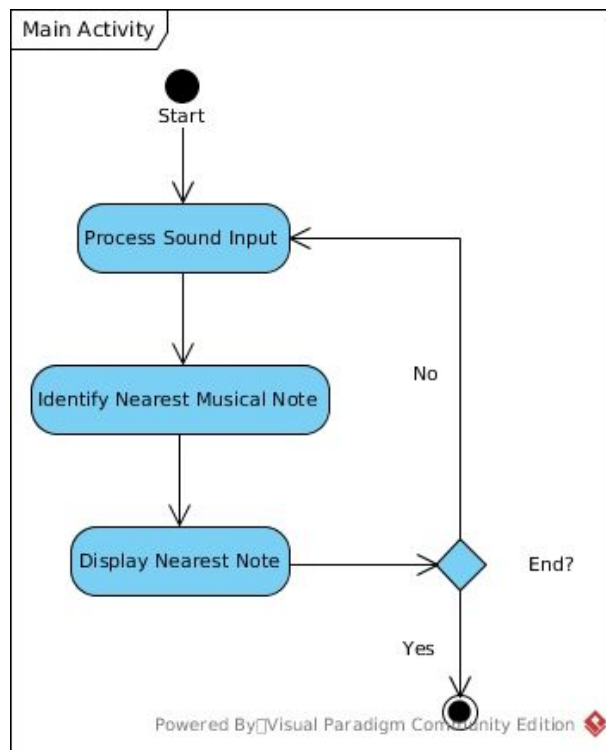
To make a guitar tuner, the sound input must be taken from the phone microphone and the frequency must be extracted and compared with the frequencies of the notes. The note closest to the played note must be displayed on the GUI.

## TEAM

Pratyay Amrit (140953430)

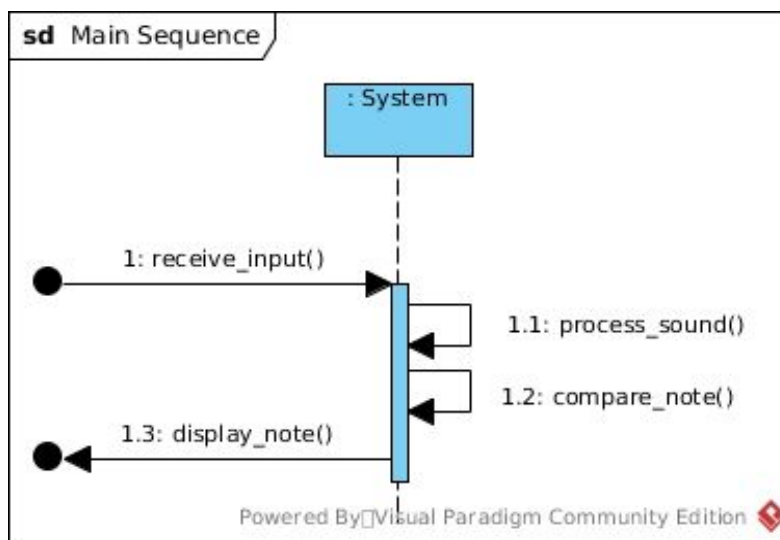
Rishabh Kanwar (140953360)

## ACTIVITY DIAGRAM



The app has a single activity. In the same, it shall respond to incoming sound signals and process it into distinct frequencies in order to compare it with all the frequencies, and identify which note is being played. The same activity will prompt the same to the user.

## SEQUENCE DIAGRAM



---

## FUNCTIONAL REQUIREMENTS

The system will need to take input from the microphone of the host device, implement the Yin Algorithm to calculate the fundamental frequency of the input signal. Compare this frequency with a hard coded array of frequency values and its corresponding musical note. These must be compared and output should be given as the nearest note to the input frequency, i.e, where the difference is minimum. Once the user has exactly reached a note, within tolerance range, the user should be prompted.

## NON-FUNCTIONAL REQUIREMENTS

The UI must be designed in a way that the user finds the app easy to use. Direct interaction with the app should be as minimum as possible, as the user will be playing a musical instrument at the same moment.

## SYSTEM REQUIREMENTS

OS: Android 4.0 Ice Cream Sandwich or above

RAM: 1GB or more

Chipset: 1.0GHz or more

## IMPLEMENTATION

To implement this application, two major Java libraries have been used. To perform the Yin Algorithm, which calculates the fundamental frequency from audio samples, a library called

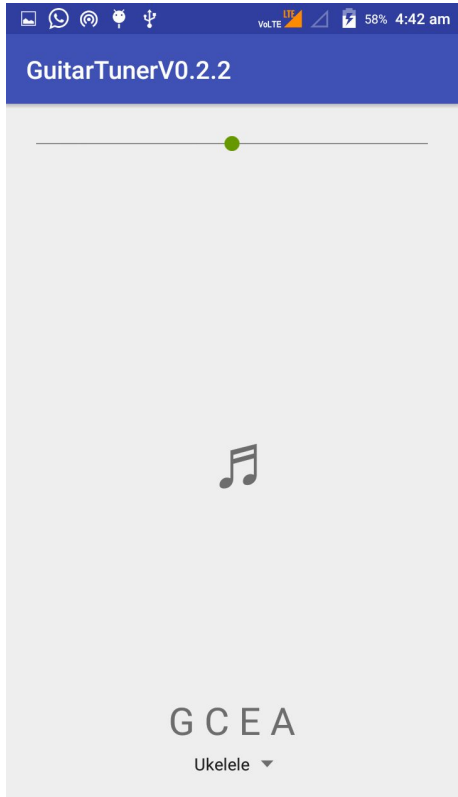
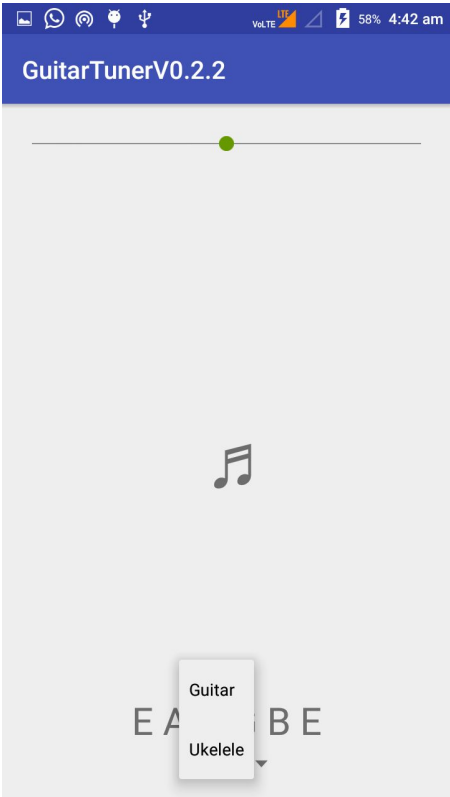
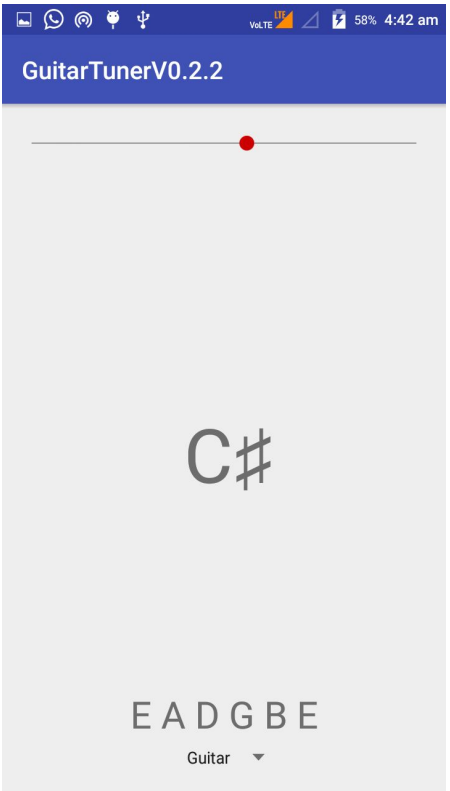
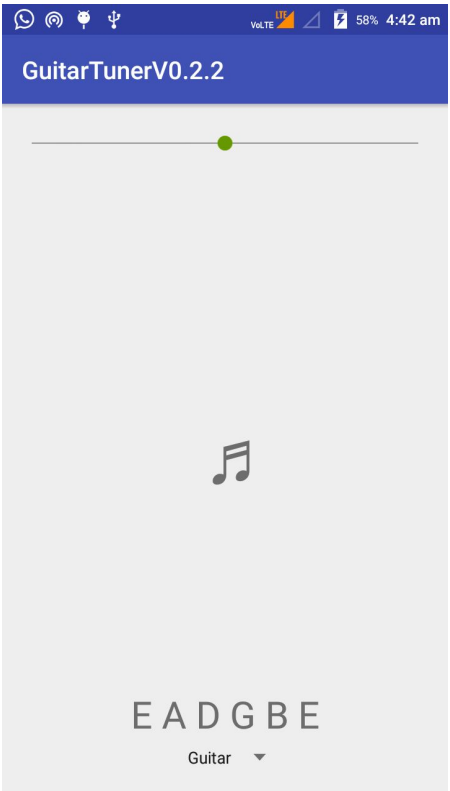
If you would like to use the project as a library, clone the project and use the desired classes. You can create an instance of a Tuner object and call it's start method when you're ready to begin listening for pitch, call it's stop method when you are finished, and remember to call it's release method when you no longer need the object. Ex:

```
Tuner tuner = new Tuner(new TunerUpdate(){
    @Override
    public void updateNote(Note newNote, PitchDetectionResult result){
        //TODO add your logic, such as View updating, here
    }
});
//To start listening for pitch
tuner.start();
//To stop listening for pitch
tuner.stop();
//When we are done with the tuner object
tuner.release();
```

Feel free to use and alter the project in accordance with the license. Also, contributions are welcome.

'TarsosDSP' by 'JorenSix' on GitHub has been used. Another library from the project 'Android-Guitar-Tuner' by GitHub user 'chRyNaN' has been used. This library translates the frequencies to their notes by performing a binary search on an indexed array of musical notes.

# SCREENSHOTS



---

## REFERENCES

- Android-Guitar-Tuner by chRyNaN (<https://github.com/chRyNaN/Android-Guitar-Tuner>)
- TarsosDSP by JorenSix (<https://github.com/JorenSix/TarsosDSP>)
- Various threads on Stack Overflow (<http://stackoverflow.com/>)
- Android Developer Reference (<https://developer.android.com/reference/classes.html>)
- TutorialsPoint (<https://www.tutorialspoint.com/>)